## Project Documentation: Flight Ticket Booking Console Application

### 1. Introduction:

The Flight Ticket Booking Console Application is a command-line based platform that enables users to search for flights, book tickets, and manage their bookings. It provides an interactive interface in the console for users to interact with the application. The project is built using Java programming language and utilizes a console-based user interface and MySQL as the database management system.

### 2. Technologies Used:

-NetBeans: NetBeans IDE is used for developing the application. It provides a user-friendly interface and powerful features for Java application development.

-Swing: Swing is a Java GUI widget toolkit used for creating the user interface of the application. It provides a set of components and layouts to design the graphical interface.

-XAMPP: XAMPP is a software package that includes Apache, MySQL, PHP, and Perl. It is used as a local development server to run the web application and manage the database.

-MySQL: MySQL is a popular open-source relational database management system. It is used to store flight details, user information, and booking records.

### 

### 3. Features and Use Cases:

**User Features:**

-Login Mode : USer can select if they want to log in as user or admin.

- Login: Users can log in to their accounts to access the booking functionality.

- Sign up: New users can create an account to register themselves in the system.

- Search Flights: Users can search for flights based on specific dates and times to find available options.

- Book Tickets: Users can select a flight from the search results and book tickets based on availability.

- My Booking: Users can view and manage their bookings, including the list of all bookings made by the user.

- Logout: Users can log out from their accounts to secure their session.

**Admin Features:**

- Admin Login: Administrators have separate login credentials to access the admin functionalities.

- Add Flights: Admins can add new flight details to the system, including flight numbers, names, source, destination, departure time, and arrival time.

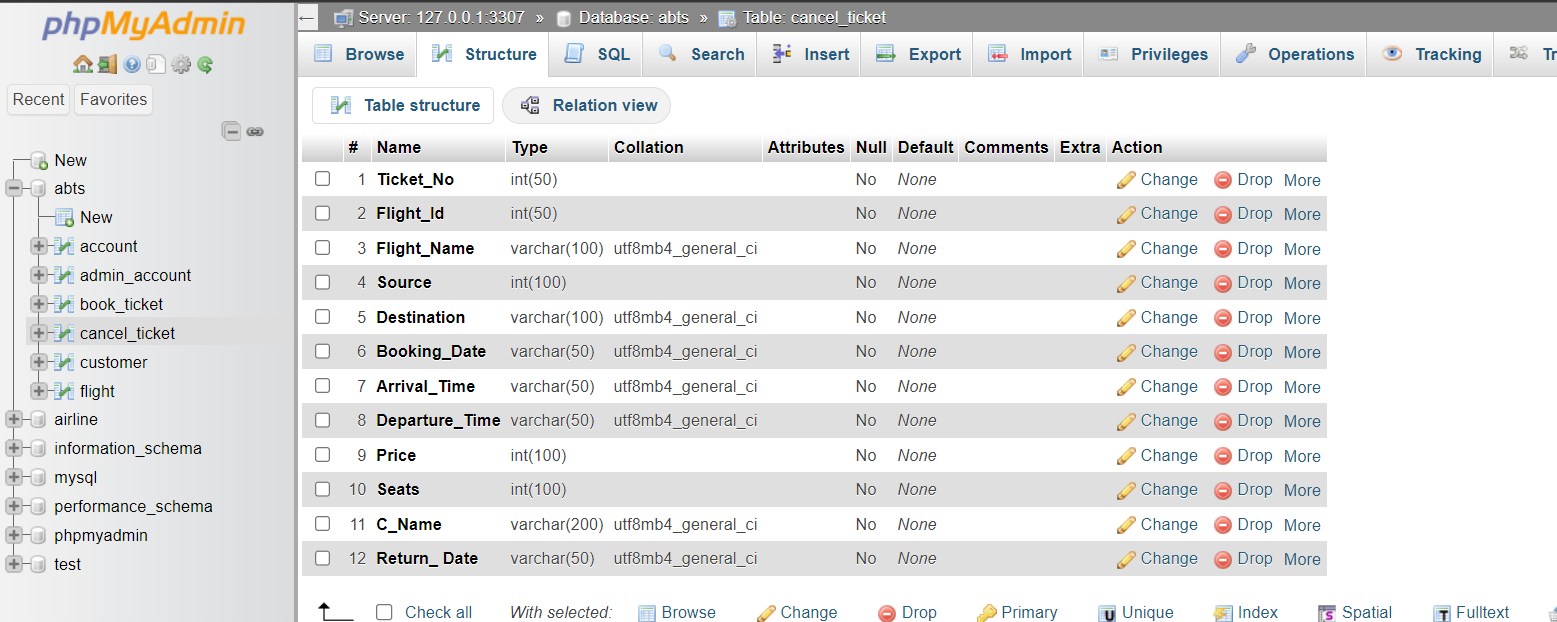
- Remove Flights: Admins can remove existing flights from the system.

- View Bookings: Admins can view bookings based on flight numbers and time.

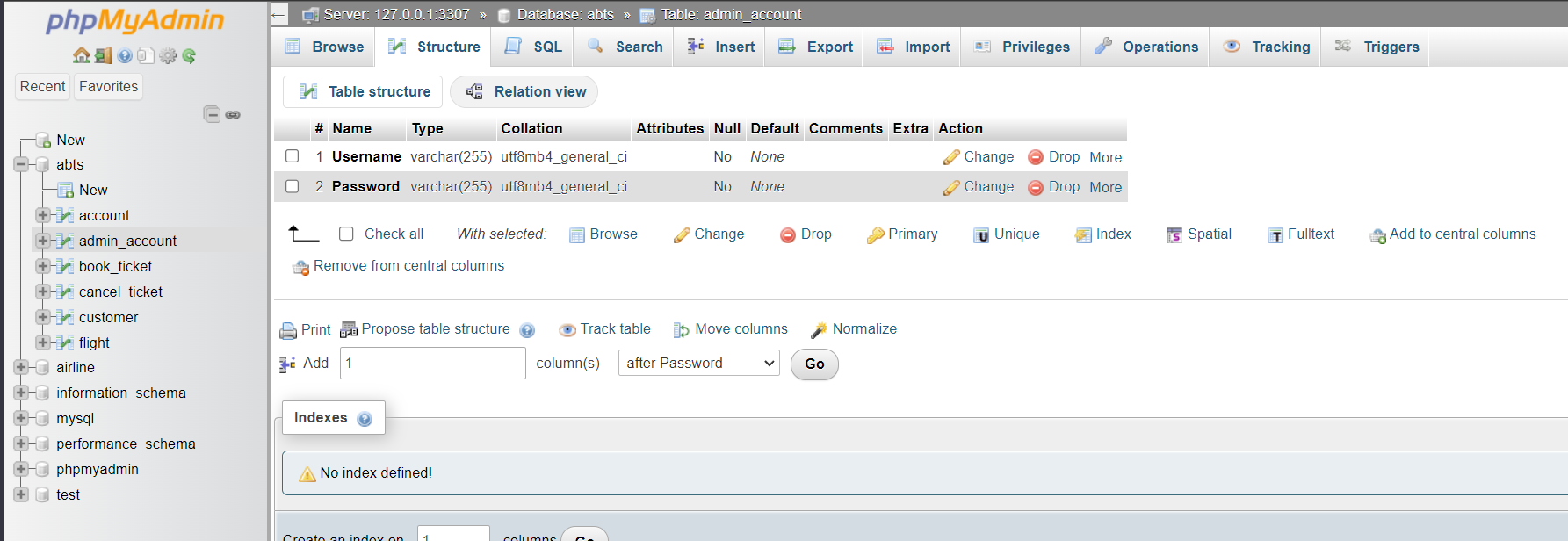
### 4. Database Design:

The MySQL database is used to store the following tables:

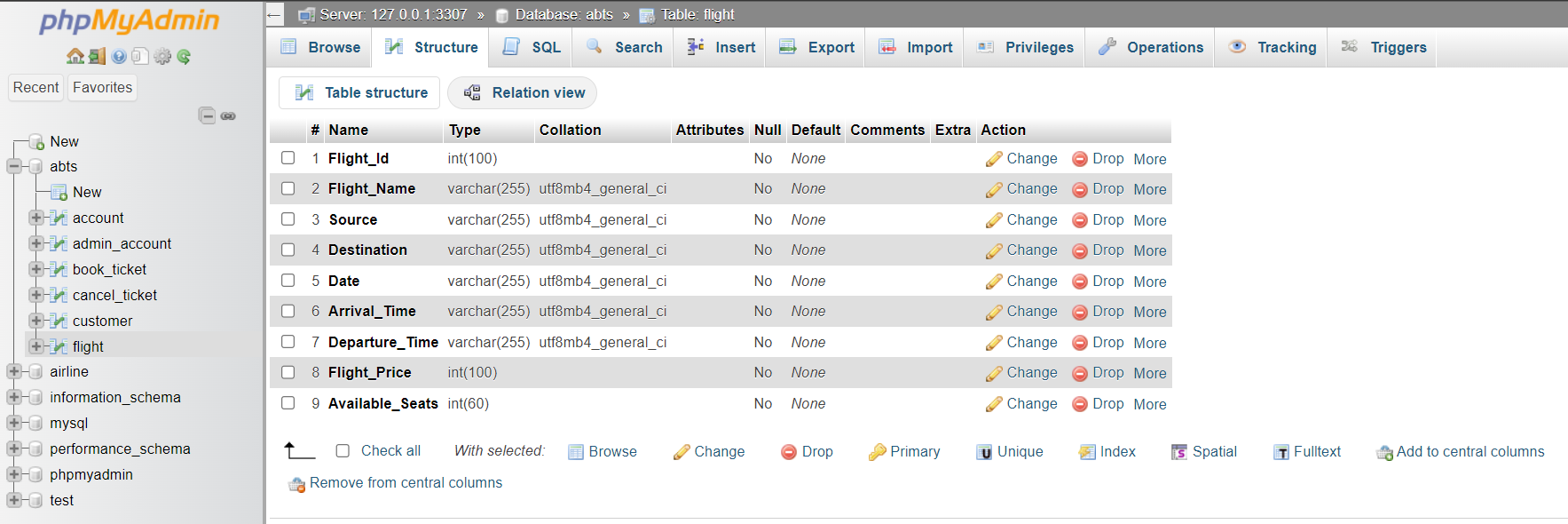
- Account: Stores user information including username, name, password, security question, and answer.



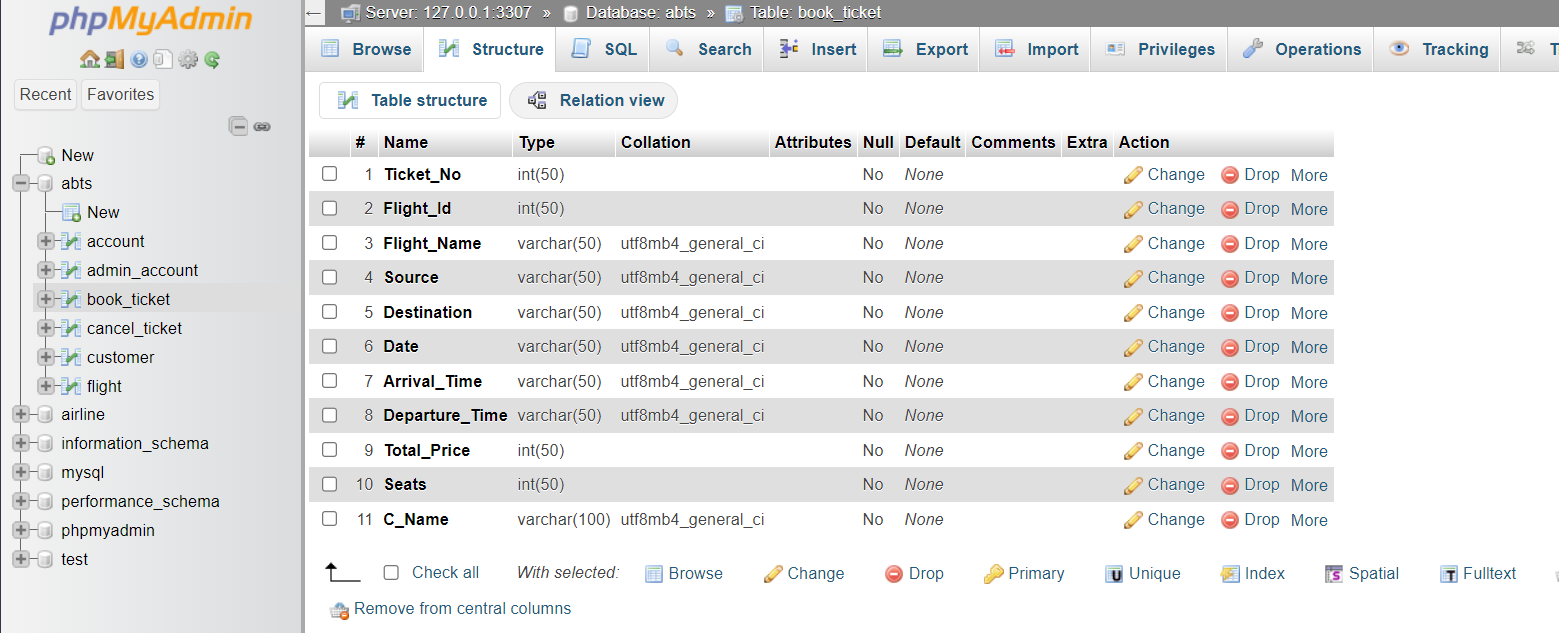
- Admin\_account : Stores Admin information such as username ,Password



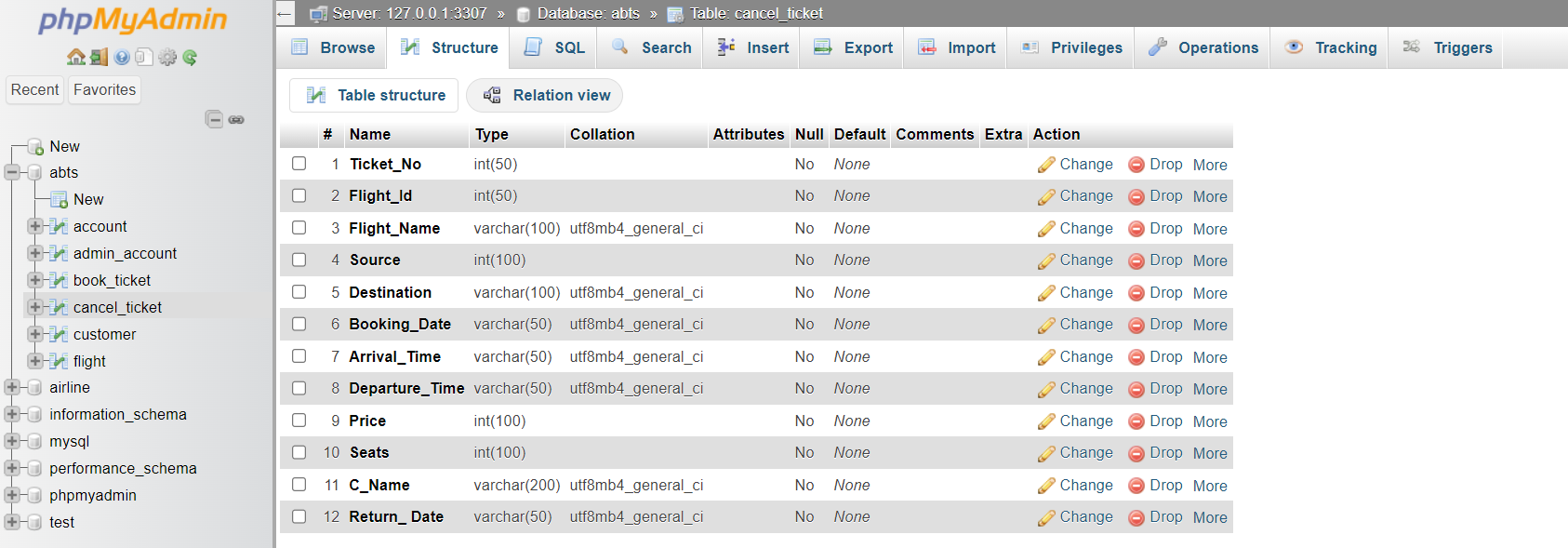
- Flight: Stores flight details such as flight ID, flight name, source, destination, departure time, and arrival time.



- Book\_table: Stores booking records with ticket numbers, flight ID, user details, date, total price, and number of seats.



-Cancel\_ticket: Stores Cancelled flight information



### 5. Installation and Configuration:

To run the Flight Ticket Booking Console Application, follow these steps:

- Install Java Development Kit (JDK) on your system.

- Install MySQL and create the necessary database and tables using the MySQL command-line or a GUI tool like phpMyAdmin.

- Import the project into an Integrated Development Environment (IDE) like Eclipse or IntelliJ.

- Configure the database connection by providing the appropriate credentials in the code.

- Build and run the application from the IDE.

### 6.Download the project

1) Download the project: Download the project files from the source (e.g., website, repository) as a compressed ZIP file.

2) Extract the downloaded file: Extract the contents of the ZIP file to a folder on your computer.

3) Open NetBeans: Launch the NetBeans IDE on your computer.

4) Create a new project: In NetBeans, go to "File" -> "New Project" to create a new project.

5) Copy all the files from the extracted folder: Open the folder where you extracted the downloaded project files. Select all the files and folders in that location.

6) Paste the files in NetBeans: In NetBeans, right-click on the "Source Packages" folder in your newly created project. Select "Paste" to copy all the files into the project's source folder.

7) Add JAR files: Right-click on the "Libraries" folder in your project, then select "Add JAR/Folder." Navigate to the folder where you stored the downloaded project files. Select the JAR files one by one and click "Open" to add them to your project's classpath.

By following these steps, you should be able to download the project, set up a new project in NetBeans, and copy the project files into the appropriate locations. Additionally, you should add the necessary JAR files to your project's classpath for any required dependencies.

Please note that it's important to review any error messages or exceptions that may occur during the process and address them accordingly.

### 7. Setting up database and running the project:

Setting up the Database:

1. Start XAMPP Server: Open XAMPP Control Panel and start the Apache and MySQL services.

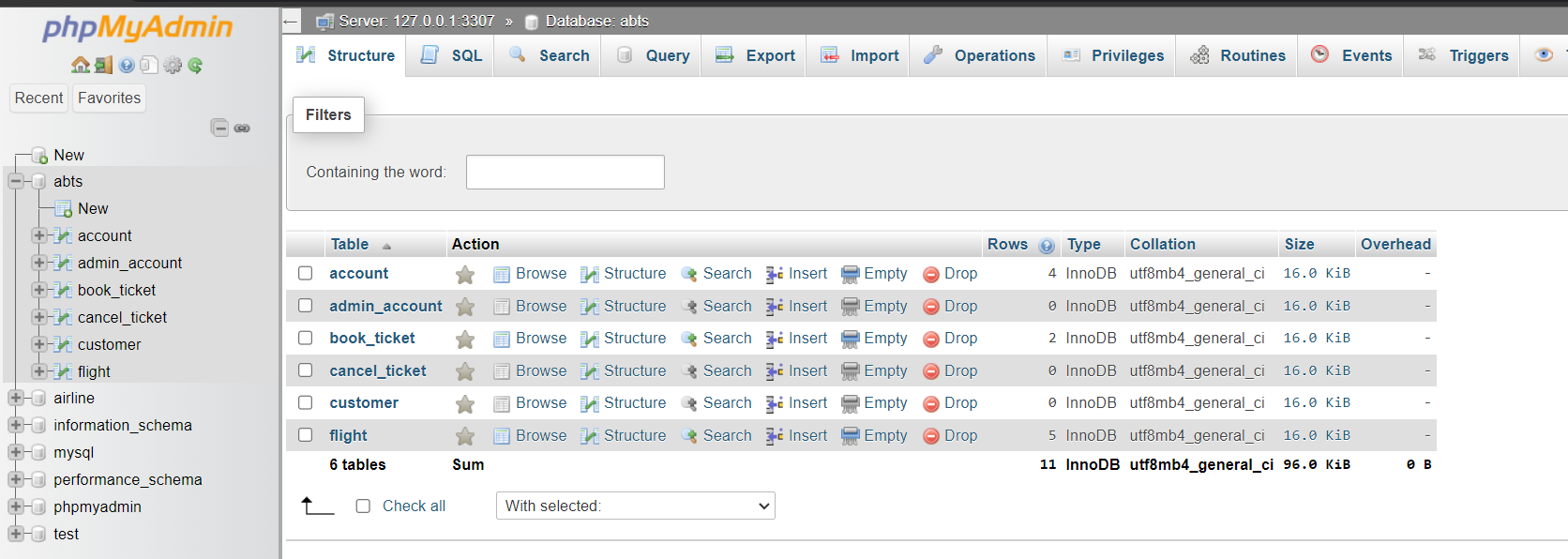
2. Access phpMyAdmin: Click on the "Admin" button for MySQL in XAMPP Control Panel. This will open phpMyAdmin in your web browser.

3. Create a New Database: In phpMyAdmin, click on the "New" tab on the left sidebar. Enter the name "atbs" for the new database and click "Create" to create the database.

4. Create Tables: Inside the newly created "atbs" database, click on the "SQL" tab on the top menu. Here, you can execute SQL queries to create tables.

5. Create Tables from SQL Scripts: Open the project in NetBeans and navigate to the "database" folder. Inside this folder, you should find SQL scripts for creating tables such as "account", "admin\_accountl", "book\_ticket", "cancel\_ticket." and "flight".

6. Execute SQL Scripts: Open each SQL script file and copy the SQL query statements. Paste the queries into the SQL tab in phpMyAdmin and click the "Go" button to execute them. Repeat this step for all the SQL scripts to create the necessary tables.



### 8.Running the Project in NetBeans:

1. Open NetBeans: Launch the NetBeans IDE.

2. Import the Project: Go to "File" -> "Open Project" and navigate to the project folder. Select the project and click "Open" to import it into NetBeans.

3. Configure the Database Connection: Open the project in NetBeans and locate the database connection configuration file or class. It may be named something like "DBConnection.java" or "DatabaseConfig.java". Inside this file, update the database connection details such as the URL, username, and password to match your MySQL setup.

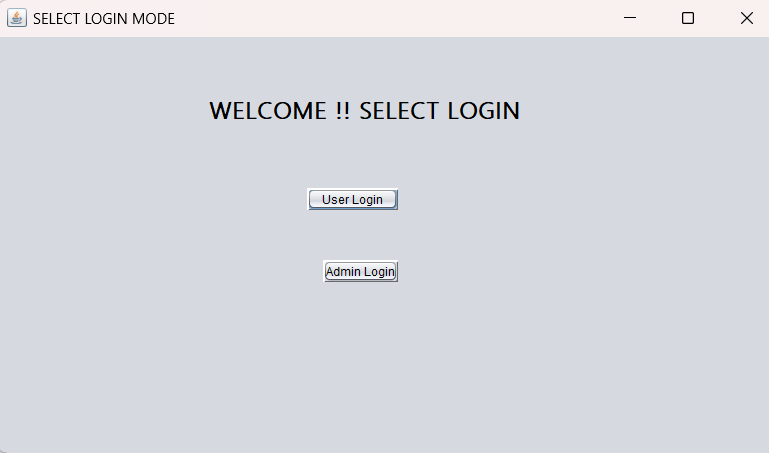
4. Build the Project: Right-click on the project in NetBeans and select "Build" or "Clean and Build" to compile the project. This will generate the executable files for the application.

5. Run the Project: Right-click on the project and select "Run" or press the "F6" key to run the project. NetBeans will execute the main class or launch the application in the console.

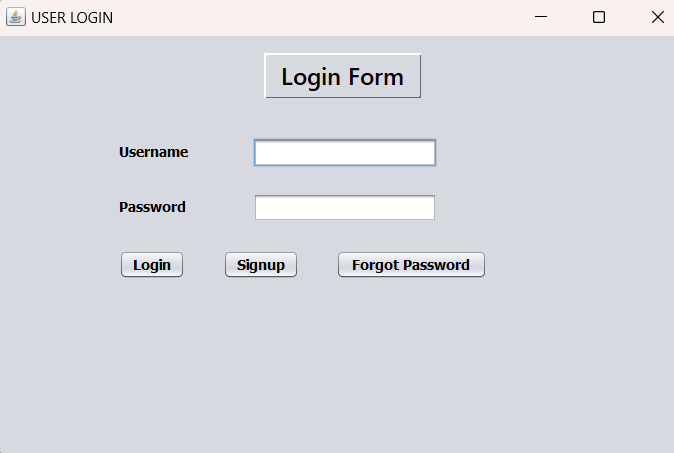
By following these steps, you should be able to set up the database, create the necessary tables, and run the Flight Ticket Booking Console Application in NetBeans. Make sure to review any error messages or exceptions that may occur during the process and troubleshoot accordingly.

### 9.Working Model:

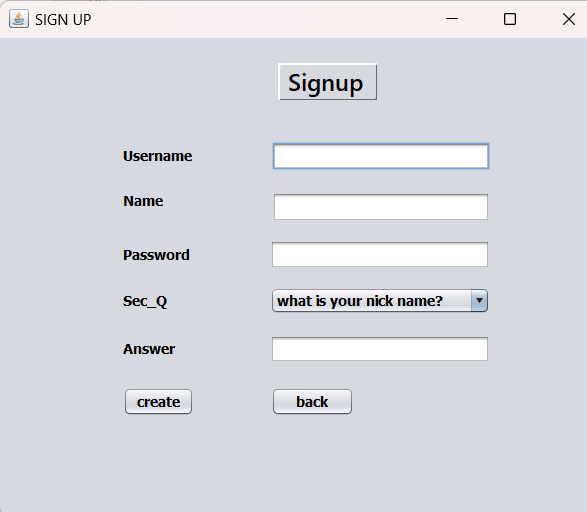
**Login Mode Page:**



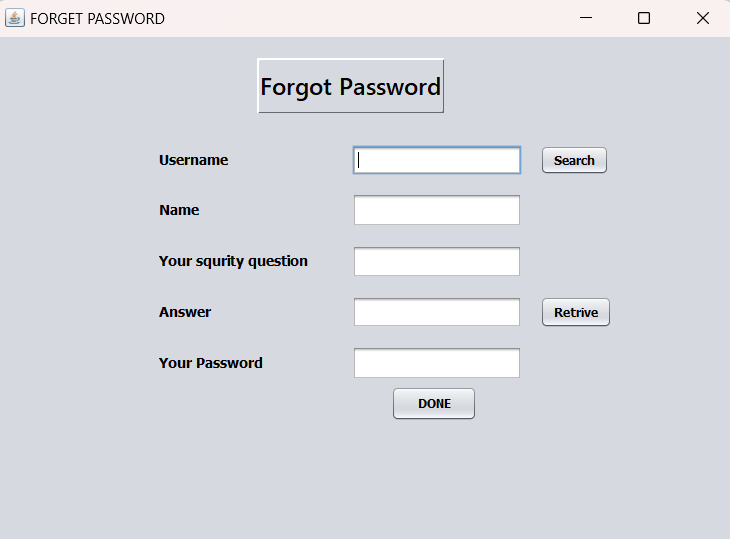
**User Login Page:**



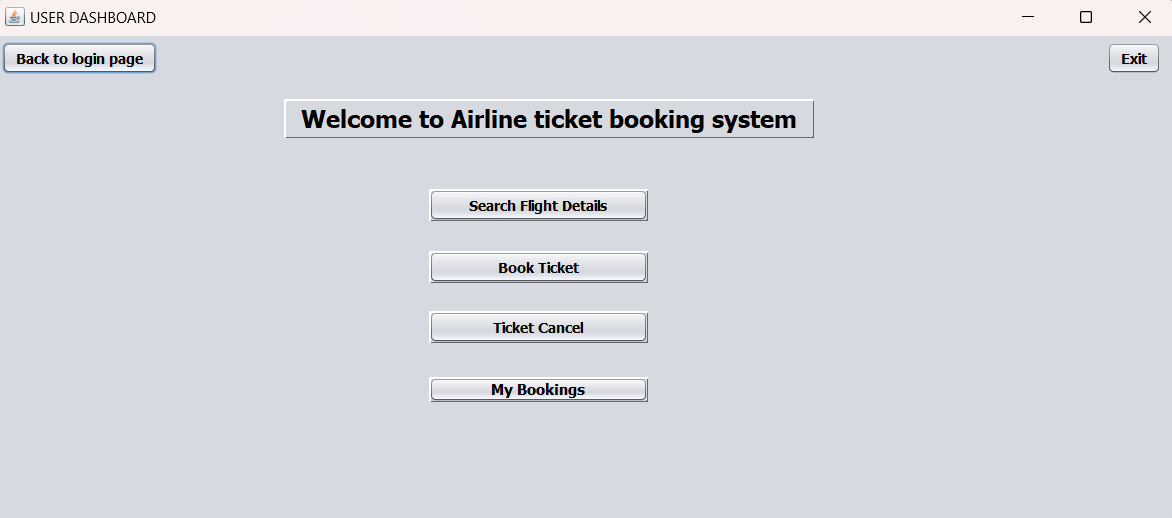
**User sign up page**:



**Forget Password :**

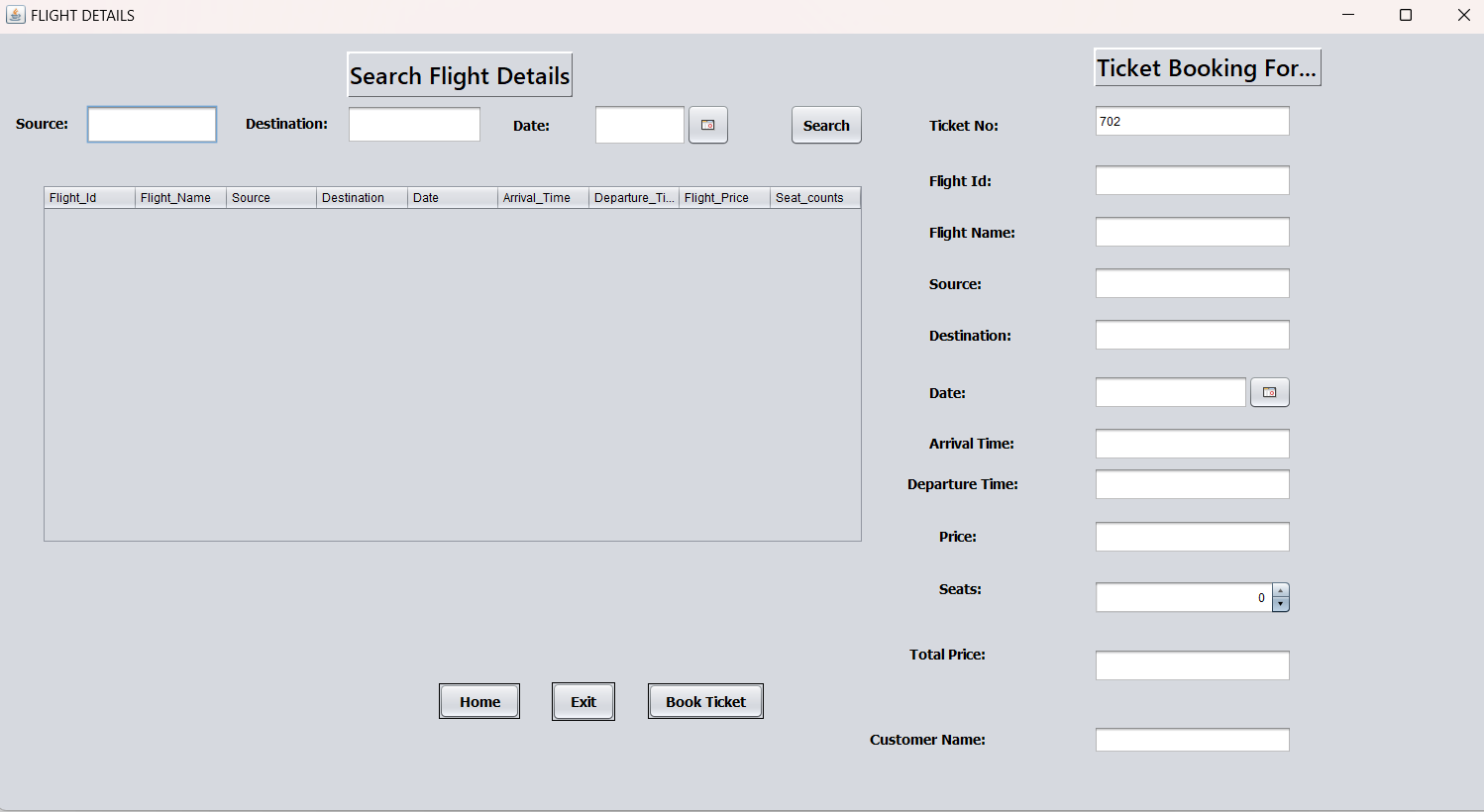


**User DashBoard:**

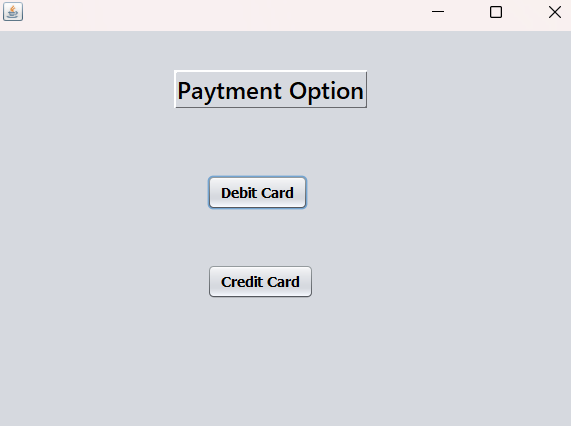


**Search Flight and Book Ticket:**

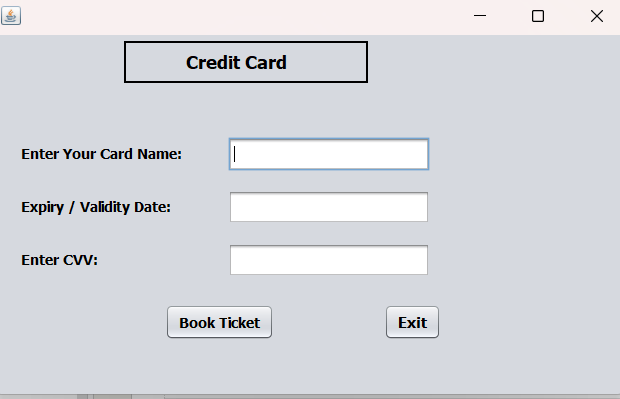
The user can search for the available flights using source and destination and date parameters and available flights are displayed in table and the user can click on the flight and it will automatically enter the details into the ticket booking form and we can specify the seat count and book the ticket using various payment methods. The booked seat count will by reduced in the flight table.



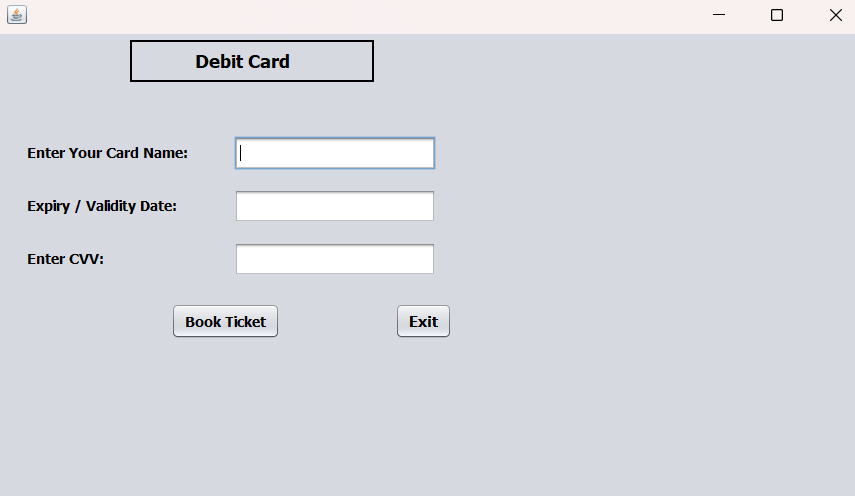
**Payment Option:**



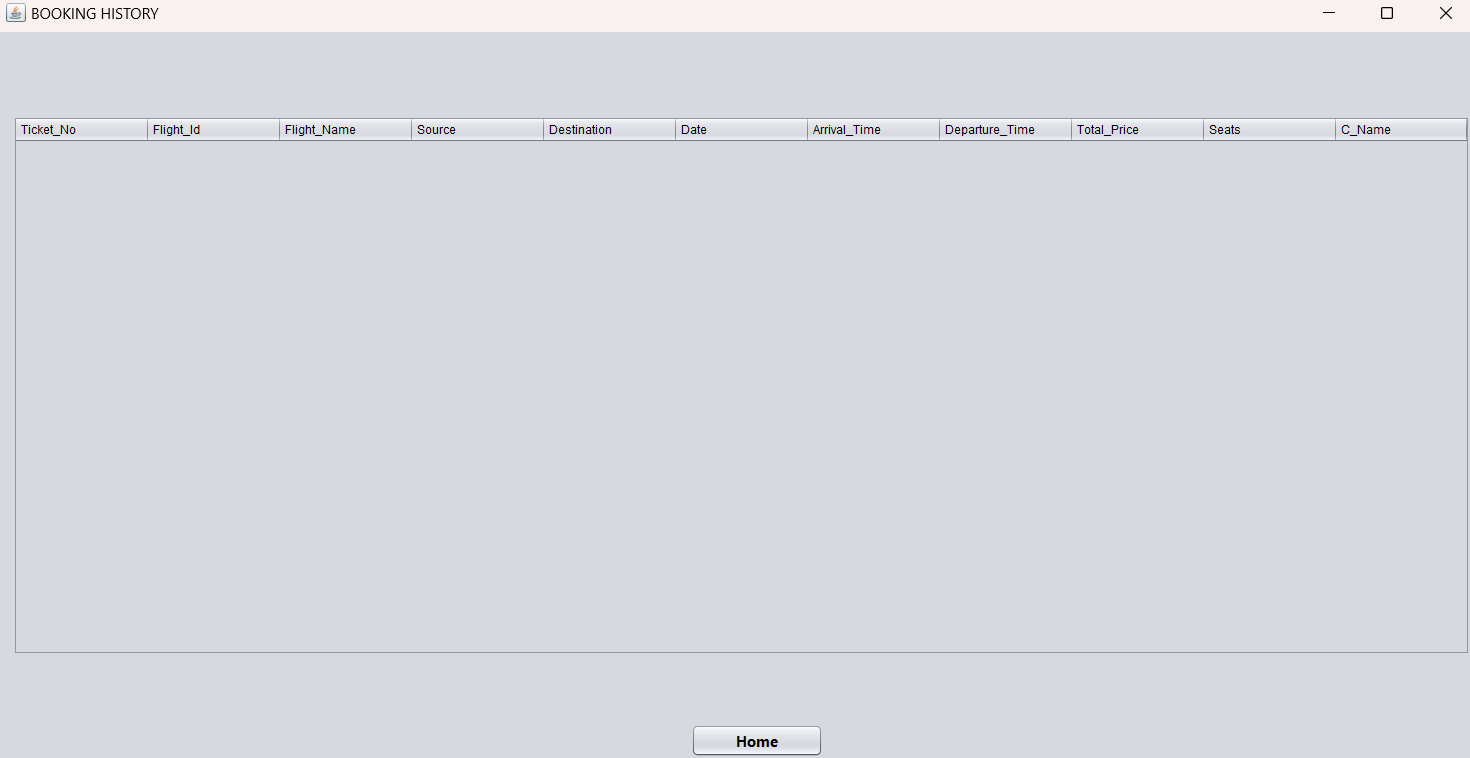
**Credit card:**



**Debit Card:**

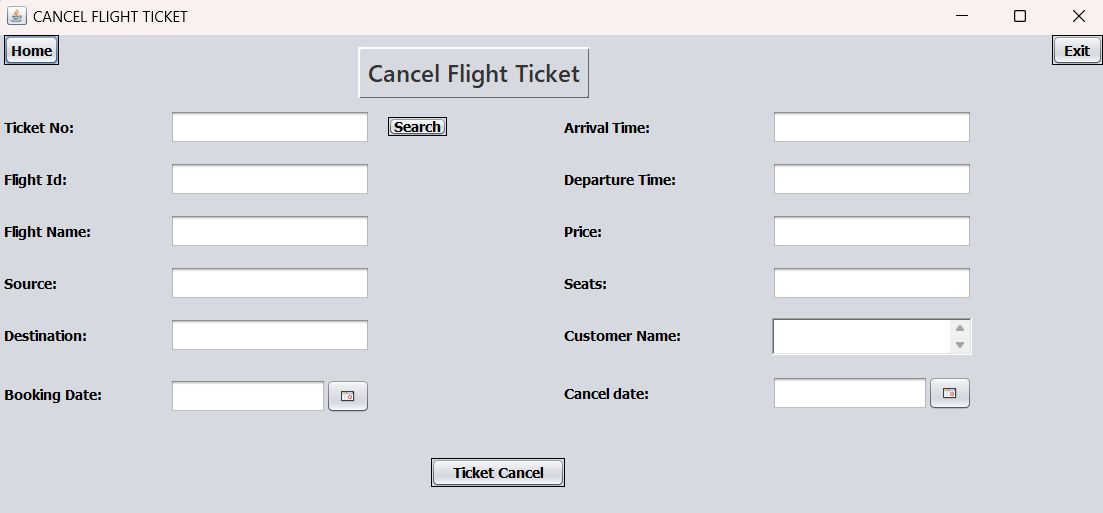
  
  
**Booking History:**

Here the user can see his booking history

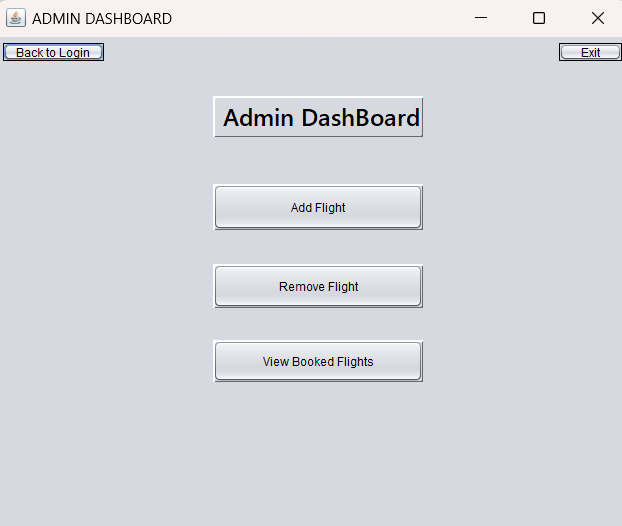


**Cancel Flight:**

Once the user decide to cancel the booking the available seats will be updated in the flight table

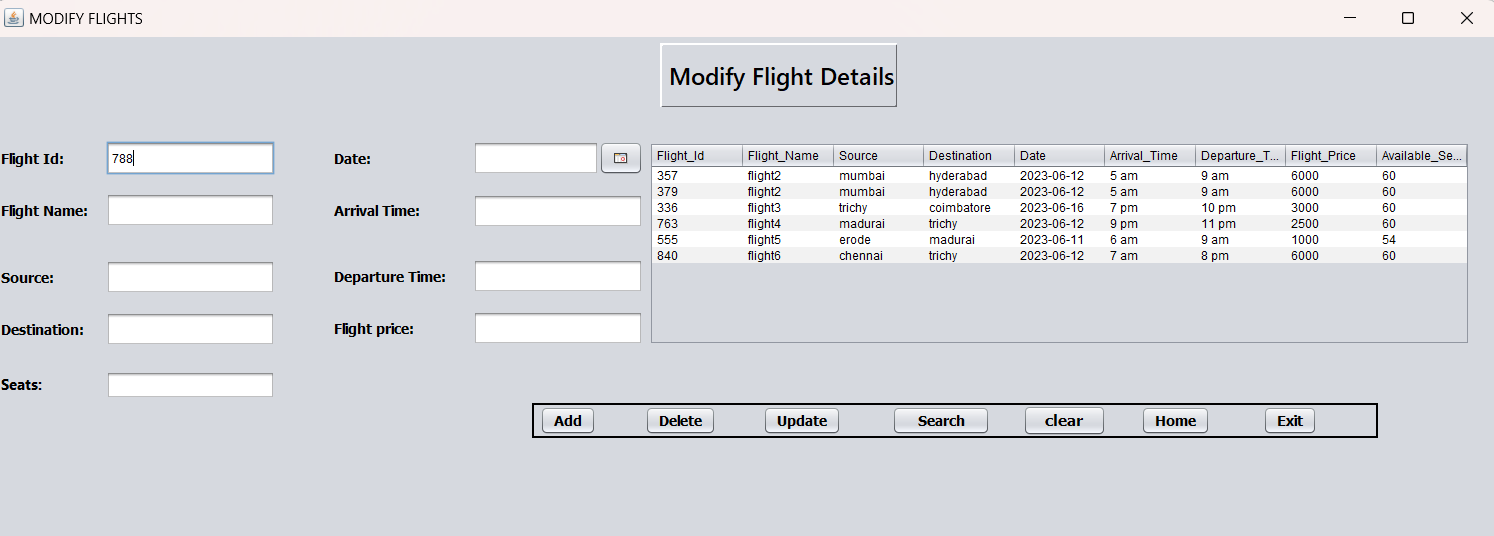


**Admin DashBoard:**



**Modify Flight Details:**

Admin can modify , Add ,Update,Delete , search the flight details



**View Booked Flights:**

Here the admin can view the overall flight bookings

